DOCUMENT RESUME

ED 057 602 EM 009 446

TITLE	Project Solo; Newsletter Number Nine.
INSTITUTION	Pittsburgh Univ., Pa. Dept. of Computer Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	22 Jan 71
NOTE	8p.; See also ED 053 566
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction; *Computer Programs; *Set Theory
IDENTIFIERS	*Project Solo

ABSTRACT

A Project Solo module which explores set theory is presented. The module allows students to define three sets by listing their elements. The student is then required to list the elements of a new set using the elements in two or more of the previous sets. The program analyzes the student's answers and tells him what, if anything, is wrong. A new version of a grading program is also listed. (JY)

PROJECT SOLO

AN EXPERIMENT IN REGIONAL COMPUTING
FOR SECONDARY SCHOOL SYSTEMS

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEH REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

University of Pittsburgh • Department of Computer Science • Pittsburgh, Pennsylvania 15213

ED057602

Newsletter No. 9

January 22, 1971

Set Theory and Computers

Most people think of computers as doing numerical computation only, and much of the application of computers to Education has been biased in this direction. Last summer, Miss Judy Richardson of Westinghouse High School developed a program which allows students to explore set theory. Since this is an area of Mathematics that does not utilize computation in the ordinary sense, there is more challenge to developing such programs than meets the eye. Judy's work is all the more remarkable when you consider that she had never used a computer before.

Inspired by this work, Frank Wimberly of our staff is developing two variations on Judy's program which he describes in the next section.

Variations on the Richardson Program

The first variation asks the student to define three sets A, B, and C by listing their elements. The student is then required to list the elements of $(A \cap B) \cup C$. The program analyzes the student's answer and tells him why it is wrong (if it is). Hence it is a tutorial program, but one based on data the student selects. The attached sample interaction gives an example of what the program looks like to a student user.

Subroutines for taking unions, intersections, and relative complements are used to compute the correct answer and to compare the correct answer with the student's answer.

The second variation being planned will permit the student to specify the operations to be performed on the sets as well as their elements. This will involve writing a parser for interpreting set theoretic expressions much like those used in computer language compilers for interpreting arithmetic expressions. The large amount of student control allowed in this variation suggests classifying it as Category II. Students and teachers are invited to try the program and to contribute to its development by supplying comments. While in NBS simply type:

```
>RUN  166FW /SETS/
```

EM 009 446

Advanced Set Theory

Progress is being made on developing an experimental module that utilizes CRS, an advanced set-theoretic information retrieval system available through our terminals. Dr. Siegfried Treu of the National Bureau of Standards in Washington is preparing this module for us, and his preliminary report looks promising. When we have had a chance to evaluate all the potential of this approach we will probably schedule a short in-service institute to explain the CRS system.

Revision of /GRADE/

A new version of 166TD /GRADE/ has been prepared. It incorporates changes suggested by Mr. Robert Gillespie. He finds his students appreciate his posting the grade bar-graph.

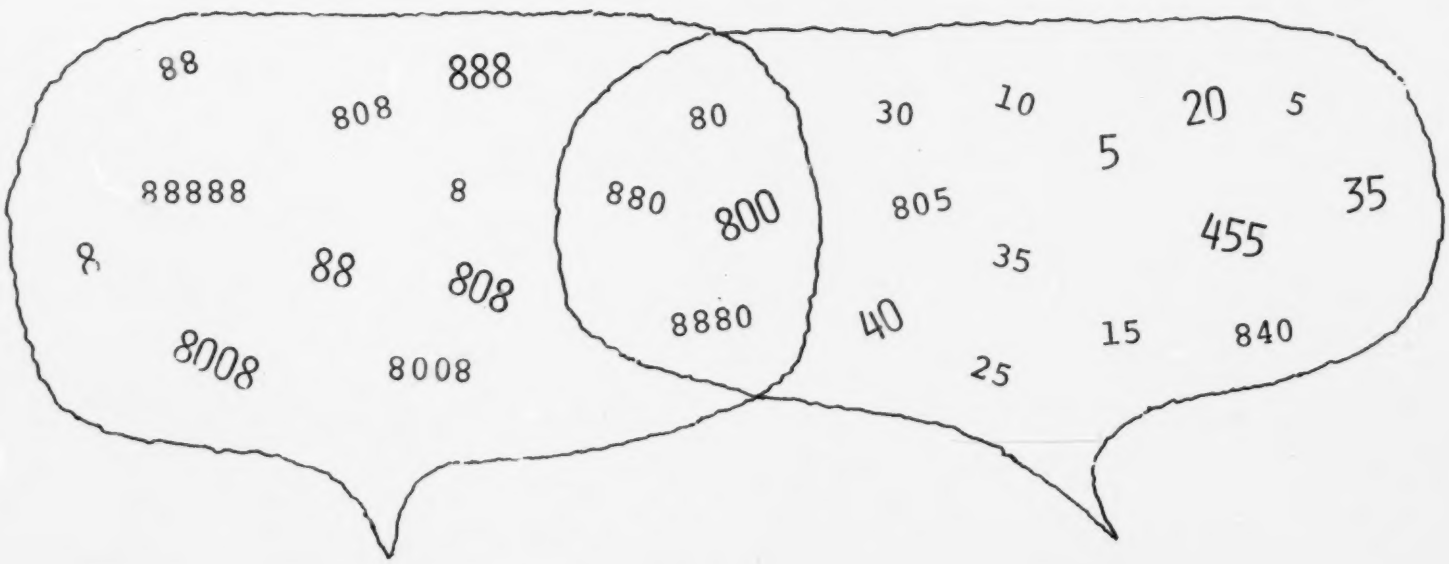
```
-CØPY 166TD /GRADE/ TØ TEL
100 VAR=ZERØ DIM P(10),G(0:100)
110 INTEGER A,T,J,J1,I,I1
120 B$=""
130 M$="      <-<-<- MEAN"
140 S$="----------I STD"
150 D$="   I"
160 E$="          I"
170 PR."WHAT IS MAXIMUM GRADE PØSSIBLE":
180 INPUT M
190 PR."ENTER AS MANY GRADES AS YØU WISH AFTER EACH '?'."
195 PR."SEPARATE GRADES WITH CØMMAS.  USE CARRIAGE RETURN"
197 PR."IF YØU NEED MØRE THAN ØNE LINE."
200 PR."ENTER 9999 WHEN FINISHED"
210 PR.
220 INPUT G(J) FØR J=1 STEP 1 UNTIL G(J-1)=9999
230 LET J=J-2
240 FØR J1=1 TØ J
250 IF G(J1)<=M THEN GØTØ 280
260 PR. "GRADE ":G(J1):" EXCEEDS MAX. INPUT ANØTHER GRADE."
270 INPUT G(J1) GØTØ 250
280 LET R1=R1+G(J1) LET R2=R2+G(J1)*G(J1)
290 LET G1=G(J1)*100/M LET S1=S1+G1 LET S2=S2+G1*G1
300 IF G1>99.9 LET G1=G1-1
310 LET G1=G1/10+1 LET P(G1)=P(G1)+1
320 NEXT J1
330 D1=SQRT((R2-(R1*R1/J))/(J-1))
340 D2=SQRT((S2-(S1*S1/J))/(J-1))
350 GØSUB 530
355 PR. PR. "RAW":TAB(5):"PERCENT"
360 PR. I:TAB(5):I
370 LET A=INT((S1/J+5)/10)*10 LET T=INT((D2+5)/10)*10
380 FØR I=1 TØ 10
385 LET I1=I*10
390 IF (I1>A-T) AND (I1<=A+T) THEN LET P$=D$ LET R$=E$ ELSE
LET P$=B$ LET R$=B$
400 IF A=I1 THEN LET R$=M$
410 IF (A-T=I1) ØR (A+T=I1) THEN LET R$=S$
420 PR. TAB(10): [PR."<*>":FØR Q=1 TØ P(I)] PR. TAB(50):P$
430 PR. INT(I*M/10):TAB(5):I1:TAB(44):R$
440 NEXT I
```

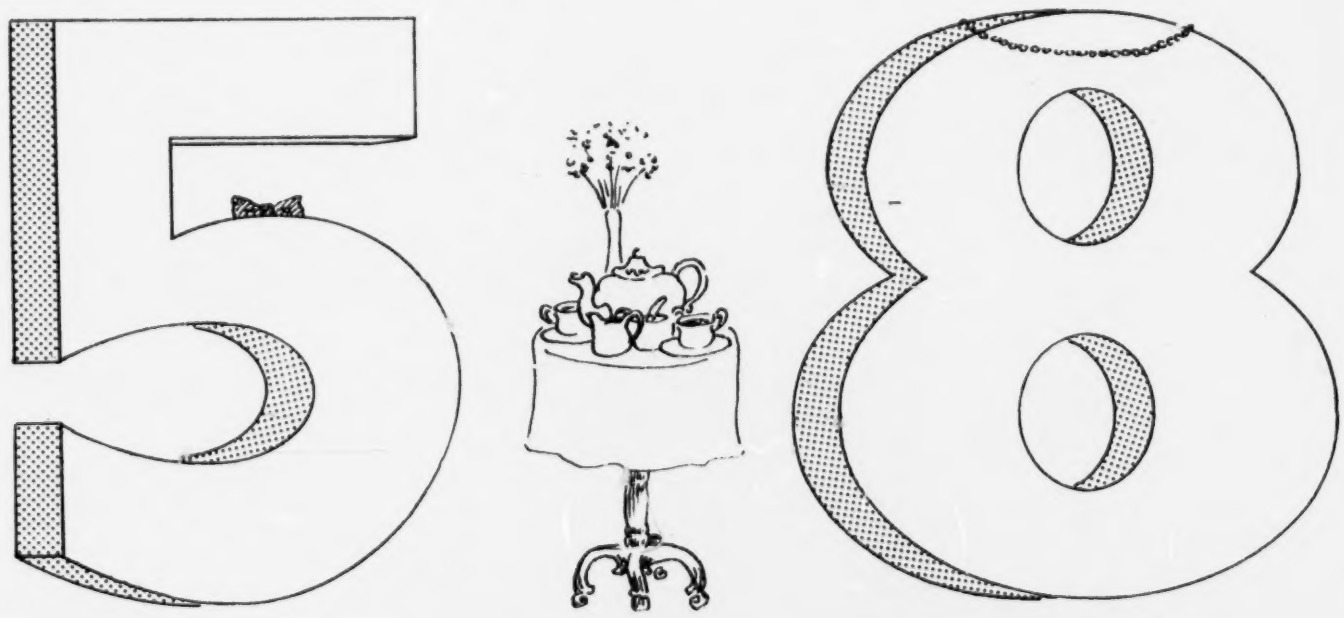
```
450 PR. GØSUB 530
460 PR. "THE UNSCALED MEAN IS ":R1/J
470 PR. "THE UNSCALED STANDARD DEVIATIØN IS ":D1
480 GØSUB 530
490 PR. "THE SCALED MEAN IS ":S1/J
500 PR. "THE SCALED STANDARD DEVIATIØN IS ":D2
510 GØSUB 530
520 END
530 PR. [PR."=-":FØR Q=1 TØ 30] PR. PR. RETURN
```

```
WHAT IS MAXIMUM GRADE PØSSIBLE?150
ENTER AS MANY GRADES AS YØU WISH AFTER EACH '?'.
SEPARATE GRADES WITH CØMMAS.  USE CARRIAGE RETURN
IF YØU NEED MØRE THAN ØNE LINE.
ENTER 9999 WHEN FINISHED

?78,98,88,56,77,45,11,198,33,111,45,66,98,100,76,55,87,86,96
?98,43,66,63,96,133,136,78,98,75,66,78,88,98,76,73,77,9999
GRADE  198 EXCEEDS MAX. INPUT ANØTHER GRADE.
?98

=-=-=-
```

INTRODUCTION TO

SET THEORY





- This module allows you to create three different sets with elements (in this case numbers) of your choice.
- The machine will then try to match wits with you in deciding which elements are in (A ∩ B) ∪ C.
- To run the program type:

```
-NBS
>RUN 166FW /SETS/
```

SAMPLE INTERACTION WITH /SETS/

```
THIS IS AN EXERCISE ØN THE ELEMENTARY ØPERATIØNS ØF
SET THEØRY--UNIØN (DENØTED BY 'U') AND INTERSECTIØN (DENØTED
BY 'I'). HENCE 'A I B' MEANS 'A INTERSECTIØN B' AND 'A U B'
MEANS 'A UNIØN B'.

AT THIS PØINT YØU MAY DEFINE THE ELEMENTS ØF THE SET
A. WHEN THE QUESTIØN MARK APPEARS TYPE AN INTEGER
NUMBER ØTHER THAN 999 (UNLESS YØU WANT A TØ BE EMPTY).
REPEAT THIS PRØCESS UNTIL A HAS AS MANY ELEMENTS AS
YØU LIKE AND THEN TYPE 999 WHEN THE NEXT QUESTIØN MARK
APPEARS. IF YØU WANT A TØ BE THE EMPTY SET TYPE 999
RIGHT AWAY.
?10
?15
?17
?22
?35
?27
?51
?999
NØW DEFINE B IN THE SAME WAY.
?15
?16
?17
?27
?38
?41
?29
?999
NØW DEFINE C.
?27
?38
?64
?37
?51
?49
?102
?999

NØW TYPE THE ELEMENTS ØF (A I B) U C. TYPE ØNE ELE-
MENT PER QUESTIØN MARK. TYPE 999 AFTER YØU HAVE TYPED THE LAST
ELEMENT. IF (A I B) U C IS EMPTY TYPE 999 RIGHT AWAY.
?27
?22
?17
?49
?10
?999
THE ELEMENTS ØF (A I B) U C ARE EITHER IN C ØR THEY
ARE IN BØTH A AND B.
YØU LEFT ØUT THE FØLLØWING ELEMENTS.
ELEMENTS ØF C.
 38     64     37     51     102
ELEMENTS ØF A I B.
  15
YØU LISTED THE FØLLØWING AS ELEMENTS ØF (A I B) U C
BUT THEY ARE NØT IN C NØR ARE THEY IN BØTH A AND B.
  22     10
WØULD YØU LIKE TØ TRY AGAIN?
?YES
```

```
NØW TYPE THE ELEMENTS ØF (A I B) U C.  TYPE ØNE ELE-
MENT PER QUESTIØN MARK.  TYPE 999 AFTER YØU HAVE TYPED THE LAST
ELEMENT.  IF (A I B) U C IS EMPTY TYPE 999 RIGHT AWAY.
?15
?17
?27
?38
?64
?37
?51
?49
?102
?999
VERY GØØD.  THAT'S EXACTLY RIGHT.
 WØULD YØU LIKE TØ TRY AGAIN?
?NØ
LISTNH
5 BASE 0
10 DIM I(20),J(20),K(20),A(20),B(20),C(20),D(20),E(20),F(20)
,G(20)
20 PR. "THIS IS AN EXERCISE ØN THE ELEMENTARY ØPERATIØNS ØF
SET THEØRY--UNIØN (DENØTED BY 'U') AND INTERSECTIØN (DENØTED
BY 'I').  HENCE 'A I B' MEANS 'A INTERSECTIØN B' AND 'A U B'
MEANS 'A UNIØN B'."
29 PR.
30 PR. "AT THIS PØINT YØU MAY DEFINE THE ELEMENTS ØF THE SET"
31 PR. "A.  WHEN THE QUESTIØN MARK APPEARS TYPE AN INTEGER "
32 PR. "NUMBER ØTHER THAN 999 (UNLESS YØU WANT A TØ BE EMPTY)."
33 PR. "REPEAT THIS PRØCESS UNTIL A HAS AS MANY ELEMENTS AS "
34 PR. "YØU LIKE AND THEN TYPE 999 WHEN THE NEXT QUESTIØN MARK"
35 PR. "APPEARS.  IF YØU WANT A TØ BE THE EMPTY SET TYPE 999"
36 PR. "RIGHT AWAY."
40 L=0
50 L=L+1
60 INPUT A(L)
70 IF A(L)<>999 THEN 50
80 A(0)=L-1
90 PRINT "NØW DEFINE B IN THE SAME WAY."
100 L=0
110 L=L+1
120 INPUT B(L)
130 IF B(L)<>999 THEN 110
140 B(0)=L-1
150 PRINT "NØW DEFINE C."
160 L=0
170 L=L+1
180 INPUT C(L)
190 IF C(L)<>999 THEN 170
200 C(0)=L-1
215 PR. "NØW TYPE THE ELEMENTS ØF (A I B) U C.  TYPE ØNE ELE-
MENT PER QUESTIØN MARK.  TYPE 999 AFTER YØU HAVE TYPED THE LAST
ELEMENT.  IF (A I B) U C IS EMPTY TYPE 999 RIGHT AWAY."
220 L=0
230 L=L+1
240 INPUT D(L)
250 IF D(L)<>999 THEN 230
260 D(0)=L-1
270 FØR L=0 TØ 20
280 I(L)=A(L)
290 J(L)=B(L)
300 NEXT L
```

```
310 GØSUB 6000
320 FØR L=0 TØ 20
330 I(L)=K(L)
340 J(L)=C(L)
350 NEXT L
360 GØSUB 5000
370 FØR L=0 TØ 20
380 E(L)=K(L)
390 NEXT L
400 FØR L=0 TØ 20
410 I(L)=E(L)
420 J(L)=D(L)
430 NEXT L
440 GØSUB 4000
450 FØR L=0 TØ K(0)
460 F(L)=K(L)
470 NEXT L
480 FØR L=0 TØ 20
490 I(L)=D(L)
500 J(L)=E(L)
510 NEXT L
520 GØSUB 4000
530 FØR L=0 TØ K(0)
540 G(L)=K(L)
550 NEXT L
560 IF F(0)<>0 THEN 590
570 IF G(0)<>0 THEN 590
580 PRINT "VERY GØØD.  THAT'S EXACTLY RIGHT."
585 GØ TØ 1000
590 PR. "THE ELEMENTS ØF (A I B) U C ARE EITHER IN C ØR THEY
ARE IN BØTH A AND B."
600 IF F(0)=0 THEN 870
610 PRINT "YØU LEFT ØUT THE FØLLØWING ELEMENTS."
620 FØR L=0 TØ 20
630 I(L)=C(L)
640 J(L)=F(L)
650 NEXT L
660 GØSUB 6000
670 IF K(0)=0 THEN 720
680 PRINT "ELEMENTS ØF C."
690 PRINT K(L); FØR L=1 TØ K(0)
720 FØR L=0 TØ 20
730 I(L)=A(L)
740 J(L)=B(L)
750 NEXT L
760 GØ SUB 6000
770 FØR L=0 TØ 20
780 I(L)=K(L)
790 J(L)=F(L)
800 NEXT L
810 GØ SUB 6000
820 IF K(0)=0 THEN 870
827 PRINT " "
830 PRINT "ELEMENTS ØF A I B."
840 PRINT K(L); FØR L=1 TØ K(0)
870 IF G(0)=0 THEN 950
880 PR. "YØU LISTED THE FØLLØWING AS ELEMENTS ØF (A I B) U C
BUT THEY ARE NØT IN C NØR ARE THEY IN BØTH A AND B."
890 PRINT G(L); FØR L=1 TØ G(0)
900 PRINT " "
950 PRINT "WØULD YØU LIKE TØ TRY AGAIN?"
```

```
960 INPUT A$
970 IF A$<>"NØ" THEN 215
980 GØ TØ 1030
1000 PRINT " WØULD YØU LIKE TØ TRY AGAIN?"
1010 INPUT A$
1020 IF A$<>"NØ" THEN 30
1030 END
4000 IF I(0)<>0 THEN 4030
4010 K(0)=0
4020 RETURN
4030 IF J(0)<>0 THEN 4080
4040 FØR L=0 TØ I(0)
4050 K(L)=I(L)
4060 NEXT L
4070 RETURN
4080 Z=0
4090 FØR L=1 TØ I(0)
4100 Y=0
4110 FØR M=1 TØ J(0)
4120 IF I(L)<>J(M) THEN 4150
4130 Y=1
4140 M=J(0)
4150 NEXT M
4160 IF Y=1 THEN 4190
4170 Z=Z+1
4180 K(Z)=I(L)
4190 NEXT L
4200 K(0)=Z
4210 RETURN
5000 IF I(0)<>0 THEN 5050
5010 FØR I1=0 TØ J(0)
5020 K(I1)=J(I1)
5030 NEXT I1
5040 RETURN
5050 IF J(0)<>0 THEN 5100
5060 FØR I1=0 TØ I(0)
5070 K(I1)=I(I1)
5080 NEXT I1
5090 RETURN
5100 I2=I(0)
5110 I(0)=I(0)+J(0)
5120 FØR I1=I2+1 TØ I(0)
5130 I(I1)=J(I1-I2)
5140 NEXT I1
5150 K(1)=I(1)
5160 K(0)=1
5170 FØR I1=2 TØ I(0)
5180 FØR I2=1 TØ I1-1
5190 IF I(I1)=I(I2) THEN 5230
5200 NEXT I2
5210 I3=K(0)+1
5215 K(0)=I3
5220 K(I3)=I(I1)
5230 NEXT I1
5240 RETURN
6000 IF I(0)<>0 AND J(0)<>0 THEN 6030
6010 K(0)=0
6020 RETURN
6030 K(0)=0
6040 FØR I1=1 TØ I(0)
6050 FØR J1=1 TØ J(0)
6060 IF I(I1)<>J(J1) THEN 6100
6070 K(0)=K(0)+1
6080 K1=K(0)
6090 K(K1)=I(I1)
6095 J1=J(0)
6100 NEXT J1
6110 NEXT I1
6120 RETURN
```